Chapter 1

Introduction

1.1 Motivation

Now a day, more and more attention is focused on content based image retrieval (CBIR) due to the tremendous growth of the number and sizes of digital image and video collections on web. So, it becomes necessary to develop power tools for retrieving this unconstrained imagery. In addition, Content Based Image Retrieval is also the key technology for improving the interface between user and computer.

Most conventional image databases are text annotated. As a result, image retrieval should be based on keyword searching. Searching using text keywords images are simple and easy to manage or manipulate. Morever, two major problems are with this method. First, creating keywords for large number of images is time consuming. Moreover, the used keywords for searching are inherently subjective and not unique. Due to these disadvantages, automatic indexing and retrieval based on image content becomes more desirable for developing large volume image retrieval applications.

Research on multimedia systems and CBIR has been given tremendous importance during last decades. The reason behind this is the fact that multimedia databases deal with large amount of text, audio, video and image data which could provide us with enormous amount of information and which has affected our life style for the better. CBIR is a bottleneck of the access of multimedia databases simply because there is a vast difference in the perception capacity between a human and computer.

One of the primary challenges in the digital libraries is the problem of providing intelligent search mechanism for multimedia collections while there are good tools for searching text collections, collections of images are much more difficult. Automated methods for searching large database of images are therefore necessary. This in turn requires effective image features for comparing images based on their overall appearance.

1.2 Problem Definition

Content Based Image Retrieval also known as query by image content and content-based visual information retrieval technique. These techniques play important role in computer vision in order to store, manage, and retrieve images data based on user query. Searching process is done with the help of matching the image features like texture, color or different combinations of them. In computer vision, image processing and pattern recognition system, texture feature of image play an important role.

It involves two steps:

a. **Feature Extraction:** First step of this process is to extract the features of images to a distinguishable extent.
b. **Matching:** Second step involves matching these features to yield a result that is visually similar.

Color has been extensively used in image matching and retrieval. But color retrieval alone does not give good results. In this report we consider the texture of the image along with the color to improve the efficiency. Color and texture features are combined in our retrieval system to compute the similar images for the given query image from the database.

1.3 Challenge

At present, more and more images are available in this digital world. Morever, this is a challenging task to find a required image for an ordinary user. There are so many researches work on image retrieval system has been carried out in the past two decades. Research in this area focuses on content based image retrieval. As processors are becoming more advance and powerful, and memories becoming cheaper, the deployment of large image databases for a different types of applications have now become realizable. Databases of art works, historical places, satellite images and medical imagery have been attracting more and more users in various professional fields--for example, advertising, geography, fashion, medicine, architecture, design, and publishing. Effectively and efficiently accessing desired images from large and varied image databases is now a necessity. In this report we are introducing a novel method of feature extraction of images using Local binary pattern

1.4 Overview of CBIR technique

The scope of this research is circumscribed to CBIR system based on color and texture features of images to improve the efficiency of the CBIR system. We have computed the image features described in Chapter 5 on mixed database and implemented color and texture feature approach on that using color and texture feature, we retrieved images from the databases for the given query image.

In many areas of web searching, security, commerce, technology, entertainment, government, and hospitals, large collections of digital images are being created. These collections are the product of digitizing existing collections of analogue paintings, photographs, drawings, diagrams, and prints. Generally, the only way of searching these collections was by browsing by keyword indexing. Digital images databases morever, open the way to content-based searching. In the content based searching contents of image are provided and similar images whose content matches with the provided content are retrieved. Specification of which images to retrieve from the database can be done in many ways. One way is to specify the image in terms of keywords used to search, or in terms of image features that are extracted from the image, such as a color histogram and another way is to browse through the database one by one. Yet another way is to provide an image or sketch from which features of the same type must be extracted as for the database images, in order to match these feature. There are several classes of features that are used to specify queries: for example color, shape, texture, faces and spatial layout. Mainly color features are often easily obtained directly from the pixel intensity, color histogram of the whole image, of a fixed sub image, or of a segmented region is often used. Yet a precise definition of texture is untraceable, the concept of texture generally refers to the presence of a spatial pattern that has some homogeneous properties. In particular, the homogeneous properties cannot result from the presence of only a single color in the region, but requires interaction of various colors. There is no universal definition of shape and the impressions of shape can be conveyed by color or intensity patterns or texture of image, from which a geometrical representation can be derived.

1.5 Field of Application

There are several systems that have been proposed in recent years in the research community of content-based information retrieval.

1.5.1 Crime prevention & security check:

Automatic face recognition systems, used by police forces. Law enforcement agencies typically maintain large archives of visual evidence of criminals, including past suspects facial photographs, fingerprints, tire treads and shoeprints.

1.5.2 Medical Diagnosis: The increasing reliance of modern medicine on diagnostic techniques such as computerized tomography, histopathology, radiology, and has resulted in an explosion in the number and importance of medical images now stored by most hospitals.

1.5.3 Intellectual Property: Registration of trademark image, where a new candidate mark is compared with existing marks to ensure that there is no risk of confusion has long been recognized as a prime application area for CBIR system. Copyright protection is also a potentially important application area. There is a growing need for copyright owners to be able to seek out and identify unauthorized copies of images, especially if they have been altered in some way.

1.5.4 The military: military application of imaging technology is probably the best developed, though least publicized images. Recognition of enemy aircraft from radar screen, using satellite photographs to identify the target, and provision of guidance systems for cruise missiles are known examples. Many of the surveillance techniques used in crime prevention could also be relevant to the military field

1.5.5 Journalism: Both newspapers and stock shot agencies maintain archive of still photographs to illustrate advertising copy or articles. These archives can often be extremely large running into hundreds of millions of images and dauntingly expensive to maintain if detailed keyword indexing is provided.

1.5.6 Fashion and interior design: Similarities can also be observed in the design process in other fields, including interior design and fashion technology. Here again, designer has to work within externally imposed constraints, such as choice of materials. This ability to search a collection of fabrics to find a particular combination of color or texture is increasingly being recognized as a useful aid to the design process.

1.5.7 Education and training: It is often difficult to identify good teaching material to illustrate key points in a lecture or self-study module. Thus, the availability of searchable collections of video clips providing examples of avalanches for a lecture on traffic congestion, or mountain safety for a course on urban planning that could reduce preparation time and lead to improved teaching quality. In some particular cases (complex diagnostic and repair procedures) such videos might even replace a human tutor.

1.5.8 Home entertainment: Home entertainment is image or video based, including holiday snapshots, home videos and scenes from favourite TV programmes or films. This is one of the few areas where a mass market for CBIR technology could develop. Possible applications could include management of family photo albums or clips from commercial films.

1.5.9 Web searching: Text based search engines have grown rapidly in usage as the Web has expanded; the well-publicized difficulty of locating images on the Web [28] indicates that there is a clear need for image search tools of similar power. Still, there is also a need for software to *prevent* access to images which are deemed pornographic.

1.6 Organization of Report:

This report is organized as follows:

General Description of the working of a conventional CBIR System is described in Chapter 2. Overview of CBIR techniques is also discussed in Chapter 2.Basics of color representation is discussed in Chapter 3. Basics of texture representation is discussed in Chapter 4.Chapter 5 describes proposed color and texture based image retrieval algorithm and computation of these features from images and then describes the experimental results of this project and finally conclusion and future work is mentioned in Chapter 6.

Chapter 2

Content Based Image Retrieval

In content-based image retrieval systems [1] (figure 2.1) the visual contents of the images in the database are extracted and described by multi-dimensional feature vectors The feature vectors of the images in the database form a feature database. For retrieval the images, users provide the retrieval system with example sketched figures or images. The system then changes these examples into its internal representation of feature vectors extracted from images. The similarities distances between the feature vectors of the query example or sketch and those of the images in the database are then calculated and retrieval is performed with the aid of an indexing scheme. This indexing scheme provides an efficient way to search for the image database. Recently, retrieval systems have incorporated users' relevance feedback to modify the retrieval process in order to generate perceptually and semantically more meaningful retrieval results.

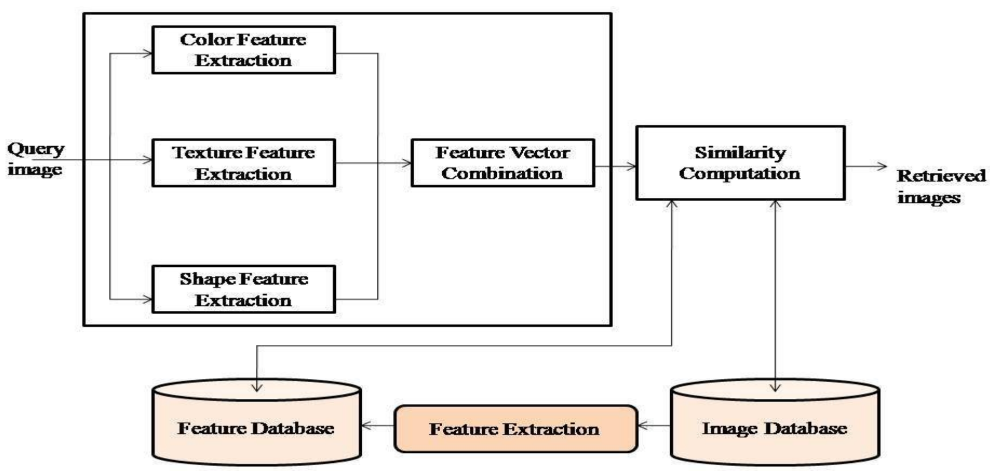


Figure 2.1: Content Based Image Retrieval System

2.1 Literature Survey

In this field, image retrieval can be traced back to the late 1970s. In 1979, a conference on Database Techniques for Pictorial Applications was held in Florence. Since then, the application potential of image database management techniques has attracted the attention of researchers. Early techniques were not generally based on visual features but on the textual annotation of images. In other words, images were first annotated with text and then searched using a text-based approach from traditional database management systems.

Text-based image retrieval uses traditional database techniques to manage images. Through text descriptions, images can be organized by topical or semantic hierarchies to facilitate easy navigation and browsing based on standard Boolean queries. Morever, since automatically generating descriptive texts for a wide spectrum of images is not feasible, most text-based image retrieval systems require manual annotation of images. Obviously, annotating images manually is a cumbersome and expensive task for large image databases, and is often subjective, context-sensitive and incomplete. As a result, it is difficult for the traditional text-based methods to support a variety of task-dependent queries.

In recent years, a drastic increase in the size of image databases has been realized. Image hosting website and online community platform claims to host more than two billion images during last few years. Due to these developments, cataloguing, annotating, and accessing effectively to these data have been significantly requested in various applications such as image search engine, grouping and filtering of Web images, biomedical information management, computer forensics, and security [22]. Therefore, many researchers from different fields of science have focused their attention on image retrieval methods. In Fig., the number of published items and citations pertaining to CBIR topic has been illustrated. CBIR as a set of challenging techniques aims to retrieve semantically requested relevant image concepts from large-scale image databases.

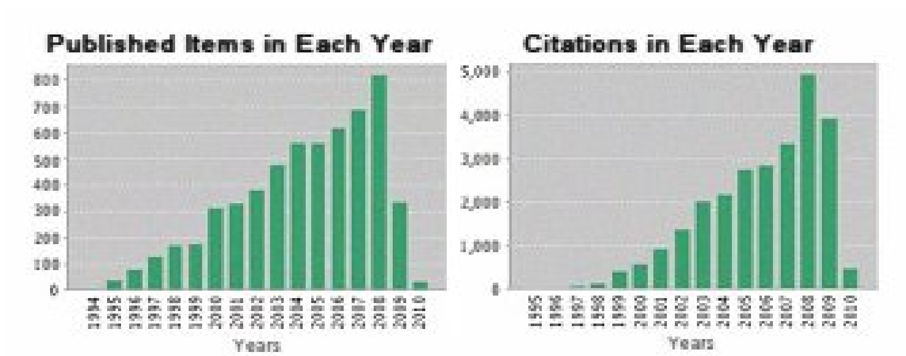


Figure: 2.2 [51]The number of published items and citations within 15 years with the topic of CBIR obtained by ISI Web of Knowledge

2.1.1 Supervised Learning Techniques

Supervised image learning is an important process to accelerate image retrieval speed, improve retrieval accuracy, and perform annotation automatically. In this off-line process, a collection of category labels and related visual features are used. Image classification can be more useful when the image training sets are well identified. Datta et al.[52] categorized the classification of images into discriminative and generative frameworks. In discriminative model, boundaries of classification are directly determined.

2.1.2 Unsupervised Learning Techniques:

Unsupervised clustering is another important technique used in content-based image retrieval. The aim of this approach is to categorize a collection of image data in a way to maximize the similarity within clusters (high intra-cluster similarity) and minimize the similarity between the clusters (low inter-cluster similarity) [21]. Datta et al. [5] divided clustering techniques into three different types in terms of image signatures: pair-wise distance based method, optimization of an overall clustering quality measure, and statistical modeling.

K-means algorithm is one of the most popular methods in clustering based on optimization quality of clusters. In this method, the centre vector of each cluster (in mass cluster representation) is employed to minimize the sum of internal-cluster distances. Li et al. [28] developed a new algorithm based on statistical modeling and optimization (D2-clustering), in which data points (region-based image signatures) characterized by a set of probability weighted vectors. D2-clustering aims to generalize k-means algorithm by using sets of weighted vectors instead of vectors. Their method for real-time automatic image annotation attempts to establish probabilistic relationships between images and relevant labels.

2.1.3 Relevance Feedback Approaches

The subjectivity of human perception is one of the key motivating reasons to make use of interaction model and specifically relevance feedback in CBIR systems. Y ong et al. [29] discussed that these sorts of techniques can help user to have high-level subjective query. As pointed out by the authors, human perception subjectivity can be appeared at the different level of subjectivity. For instance, people under different circumstances may recognize the same image content in a different way. Continuous (cumulative) learning is other motivating factor to employ these kinds of techniques [21]. To develop an efficient algorithm to reveal user's preferences is the main goal of interactive techniques.

Existing CBIR system

Several systems have been proposed in recent years in the research and community for content based information retrieval:

a. **QBIC** or **Query by Image Content [12]** was developed by IBM, Almaden Research Centre, to allow users to graphically pose and refine queries based on multiple visual properties such as color, texture and shape [4]. It supports queries based on input images, user-constructed sketches, and selected color and texture patterns.

b. **VIR Image Engine** by Virage Inc., like QBIC, enables image retrieval based on primitive attributes such as color, texture and structure. It examines the pixels in the image and performs an analysis process, deriving image characterization features [12].

c. **Visual SEEK** and **Web SEEK** were developed by the Department of Electrical Engineering, Columbia University. Both these systems support color and spatial location matching as well as texture matching [12].

d. **NeTra** was developed by the Department of Electrical and Computer Engineering, University of California. It supports color, shape, spatial layout and texture matching, as well as image segmentation [12], [7].

e. **MARS** or **Multimedia Analysis and Retrieval System** was developed by the Beckman Institute for Advanced Science and Technology, University of Illinois. It supports color, spatial layout, texture and shape matching [12].

2.2 Content Comparison Techniques

There are some common methods for extracting content from images so that they can be easily compared. The methods outlined are not specific to any particular application domain.

2.2.1 Color Retrieval

Color is the most extensively used visual content for image retrieval. Its three dimensional values make its discrimination potentiality superior to the single dimensional gray values of images. Before selecting an appropriate color description, color space [3] must be determined first. Retrieving images based on color similarity is achieved by computing a color histogram for each image that identifies the proportion of pixels within an image holding specific values. The first order (mean), the second order (variance) and the third order (skewness) color moments have been proved to be efficient and effective in representing color distributions of images.

A different way of incorporating spatial information into the color histogram, color coherence vectors (CCV), was proposed. Each histogram bin is partitioned into two types, i.e., coherent, if it belongs to a large uniformly-colored region, or incoherent, if it does not. Another method called color correlogram expresses how the spatial correlation of pairs of colors changes with distance.

2.2.2 Texture Retrieval

Texture is a widely used and intuitively obvious but has no precise definition due to its wide variability. Visual texture [4] in most cases is defined as a repetitive arrangement of some basic pattern. This repetition may not be random. Morever, a texture pattern normally has some degree of randomness due to randomness in basic pattern as well as due to randomness in the repetition of basic pattern. To quantify texture, this randomness is measured by some means over a small rectangular region called window. Thus, texture in an image turns out to be a local property and depends on the shape and size of the window. Identifying a patch in an image as having uniform texture or discriminating different visual textures obeys the law of similarity. In this case, the texture property is used to produce similarity groupings.

Basically, texture representation methods can be classified into two categories: *structural* and *statistical*. Structural methods, including *morphological operator* and *adjacency graph*, describe texture by identifying structural primitives and their placement rules. They tend to be most effective when applied to textures that are very regular. Statistical methods, including *Fourier power spectra* [5], *co-occurrence matrices*, *shift-invariant principal component analysis* (*SPCA*) [7], *Tamura feature*, *Wold decomposition*, *Markov random field*, *fractal model*, and *multi-resolution filtering* techniques such as *Gabor and wavelet transform*, *local binary pattern,* characterize texture by the statistical distribution of the image intensity.

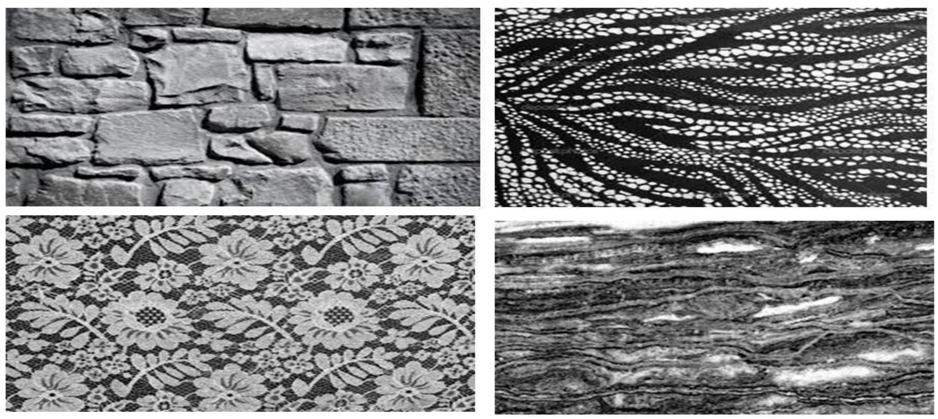


Figure 2.3: Different types of texture

2.3 Similarity/Distance Measure

Instead of exact matching, content-based image retrieval calculates visual similarities between a query image and images in a database. Accordingly, the retrieval result is not a single image but a list of images ranked by their similarities with the query image. Many similarity measures have been developed for image retrieval based on empirical estimates of the distribution of features in recent years. Different ***similarity/distance measures*** will affect retrieval performances of an image retrieval system significantly. In this section, we will introduce some commonly used similarity measures. We denote D (I, J) as the distance measure between the query image I and the image J in the database; and fi (I) as the number of pixels in bin i of I.

2.3.1 Minkowski-Form Distance

If each dimension of image feature vector is independent of each other and is of equal importance, the *Minkowski-form distance* L_n is appropriate for calculating the distance between two images. This distance is defined as:

The Minkowski distance of order n between two points$P = (x_1, x_2, x_3, \dots \dots \dots \dots x_n)$ and $Q = (y_1, y_2, y_3, \dots \dots \dots \dots y_n) \in \mathbb{R}^n$

is defined as:

$$D(I, J) = (\sum_{i=1}^{n} |f_i(I) - f_i(J)|^{n})^{\frac{1}{n}}$$

When n=1, 2,………∞, *D* (*I*, *J*) is the *L1*, *L2* (also called Euclidean distance), and *L*∞ distance respectively. Minkowski-form distance is the most widely used metric for image retrieval .Netra [7] used Euclidean distance for color and shape feature, and *L1* distance for texture feature

The **Histogram intersection** can be taken as a special case of *L1* distance, which is used by Swain and Ballard [8] to compute the similarity between color images. The Intersection of the two histograms of *I* and *J* is defined as

$$S(I, J) = \frac{\sum_{i=1}^{N} min\,(f_i(I), f_i(J))}{\sum_{i=1}^{N} f_i(J)}$$

It has been shown that histogram intersection is fairly insensitive to changes in image resolution, histogram size, occlusion, depth, and viewing point.

2.3.3 K-Means Classifier

The k-means algorithm [11] is a fast iterative algorithm that has been used in many clustering applications. It finds locally optimal solutions with respect to the clustering error. Let us consider a data set X= $\{ x_1, x_2, x_3, \dots \dots \dots \dots x_n \}$ X $\in R^d$where d is the dimension. The K-clustering problem aims at partitioning this data set into K independent clusters $\{C_{1,} C_2, C_3, \dots \dots \dots \dots C_k\}$, such that squared Euclidean distances between each data point x_i and the centroid m_c (cluster center) of the subset C_K is minimized. This criterion is called clustering error and depends on the cluster centers $\{ m_1 m_2 m_3 \dots \dots \dots .. m_K \}$

E $(m_1 m_2 m_3 \dots \dots \dots .. m_K) = \sum_{i=1}^{M} \sum_{j=1}^{N} I\left(x_i \in C_j\right) ||x_i - m_j||^2$

Where I(X) =1 if X is true and 0 otherwise.

2.4 Query techniques

Different implementations of CBIR make use of different types of user queries.

2.4.1Query by example

In this technique a user provides example image to the CBIR system for which it wants to search similar images .In this case search is based on some common attributes that provided image sharing with the searched image. There are many ways by which a user can provide query image:

- An image already in the database can be supplied by the user or it can chose from a random set.
- The user can draw a rough approximation of the image they are looking for.

This query technique removes the difficulties that can arise when trying to describe images with words.

2.4.2 Semantic retrieval

The ideal CBIR system from a user perspective would involve what is referred to a semantic retrieval, where the user makes a request like "find pictures of lion" or even "find pictures of Abraham Lincoln". This type of open-ended task of searching is very difficult for computers to perform - pictures of Chihuahuas and Great Danes look very different, and Lincoln may not always be facing the camera or in the same pose. Current CBIR systems therefore generally make use of lower-level features like texture, color, and shape, yet some systems take advantage of very common higher-level features like faces). Not every CBIR system is generic.

2.4.3 Other query methods

Other query methods include browsing for example images, navigating customized/hierarchical categories, querying by image region (rather than the entire image), querying by multiple example images, querying by visual sketch, querying by direct specification of image features, and multimodal queries (e.g. combining touch, voice, etc.). CBIR systems can also make use of relevance feedback, where the user progressively refines the search results by marking images in the results as "relevant", "not relevant", or "neutral" to the search query, then repeating the search with the new information.

Chapter 3
Color Representation

Color is the brains reaction to a specific visual stimulus. Yet we can precisely describe color by measuring its spectral power distribution the intensity of the visible electro-magnetic radiation at many discrete wavelengths this leads to a large degree of redundancy. Main reason for this redundancy is that the eye's retina samples color using only three broad bands, red, green and blue light. Signals from these color sensitive, together with those from the rods sensitive to intensity only, are combined in the brain to give several different sensations the color [3].

- Brightness - the human sensation by which an area exhibits more or less light.
- Hue - the human sensation according to which an area appears to be similar to one, or to proportions of two, of the perceived colors red, yellow, green and blue.
- Colorfulness - the human sensation according to which an area appears to exhibit more or less of its hue.
- Lightness - the sensation of an area's brightness relative to a reference white in the scene.
- Chroma - the colorfulness of an area relative to the brightness of a reference white.
- Saturation - the colorfulness of an area relative to its brightness.

The tri-chromatic theory describes the way three separate lights: red, green and blue, can match any visible color based on the eye's use of three color sensitive sensors. These are the basis on which photography and printing operation works, using three different colored dyes to reproduce color in a scene. It is also the way that most computer color spaces operate, using three parameters to define a color.

3.1 Color Spaces

A color space [12] is a method by which we can specify, create and visualize color. As humans, we may define a color by its attributes of brightness, hue and colorfulness. A computer may describe a color using the amounts of red, green and blue phosphor emission required to match a color. A printing press may produce a specific color in terms of the reflectance and absorbance of cyan, magenta, yellow and black inks on the printing paper. A color is thus generally specified using three co-ordinates, or parameters. These parameters describe the position of the color within the color space being used. They do not tell us what the color is, that depends on what color space is being used.

One can describe color spaces using the concept perceptual uniformity. Perceptually uniform means that two color spaces that are equally distant in the color space are perceptually equally distant. Perceptual uniformity is a very important concept when a color space is quantized. When a color space is perceptually uniform, there is less chance that the difference in color value due to the quantization will be noticeable on a display or on a hard copy.

3.1.1 HSx color space

The HSV, HSI, HSL ,HSB color spaces (HSx) are more closely related to human color perception than the RGB color space, but are still not perceptually uniform. The axes from the HSx color spaces represent hue, saturation and lightness (also called value, brightness and intensity) color characteristics [3].The difference between the different HSx color spaces is their transformation from the RGB color space. They are generally represented by different shapes (e.g. cone, cylinder). Hue describes the actual wavelength of a color by representing the color's name, for example, green, red or blue. Saturation is a measure of the purity of a color. For instance, the color red is a 100% saturated color, but pink is a low saturation color due to amount o white in it. Intensity indicates the lightness of a color. It ranges from black to white.

3.1.2 RGB color space

The RGB Color space is the most widely used color space for computer graphics. Note that R,G,B stand here for intensities of the red, green, blue guns in the CRT, not for primaries as meant in CIE RGB space. It is an additive color space: red, green and blue light are combined to create other colors. It is not perceptually uniform. The RGB color space can be visualized as cube.
The RGB color space is the prevalent choice for digital images [3] because color-CRTs (computer display) use red, green and blue phosphors to create the desired color. Also, it is easy for programmers to understand and program since this color space has been

widely used for a number of years. Morever, a major drawback of the RGB space is that it is senseless. The user finds it difficult to understand or get a sense of what a color R=100, G=50 and B=80 is and the difference between R=100, G=50 and B=50 and R=100, G=150 and B=150.

Each color-axis (R, G, B) is equally important. Therefore, each axis should be quantized with the same precision. So, when the RGB color space is quantized, the number of bins should always be a cube of an integer.

3.2 Color conversion

In order to use a good color space for a specific application, color conversion is needed between color spaces. Color translation, or color space conversion, is the translation of the representation of a color from one color space to another. This calculation is required whenever data is exchanged inside a color-managed chain. Transforming profiled color information to different output devices is achieved by referencing the profile data into a standard color space. It is easy to convert colors from one device to a selected standard and from that color space to the colors of another device. By ensuring that the reference color space covers the many possible colors that humans can see, this concept allows one to exchange colors between many different color output devices. The good color space for image retrieval system should preserve the perceived color differences. In other words, the numerical Euclidean difference should approximate the human perceived.

3.3 Color quantization

Many people don't have full-color (24 bit per pixel) display hardware. Inexpensive display hardware stores 8 bits per pixel, so it can display at most 256 distinct colors at a time. To display a full-color image, the computer must choose an appropriate set of representative colors and map the image into these colors. This process is called "color quantization”. So, **color quantization** or **color image quantization** is a process that reduces the number of distinct colors used in an image, generally with the intention that the new image should be as visually similar as possible to the original image. Clearly, color quantization is a lossy process. Since JPEG is a full-color format, displaying a color JPEG image on 8-bit-or-less hardware requires color quantization. The speed and image quality of a JPEG viewer running on such hardware is largely determined by its quantization algorithm. On the other hand, a GIF image has already been quantized to 256 or fewer colors.

3.4 Image

A digital image is composed of *pixels* which can be thought of as small dots on the screen. A digital image is an instruction of how to color each pixel. A typical size of an image is 512-by-512 pixels. In the general case we say that an image is of size *m*-by-*n* if it is composed of *m* pixels in the vertical direction and *n* pixels in the horizontal direction.

Let us say that we have an image on the format 512-by-1024 pixels. This means that the data for the image must contain information about 524288 pixels, which requires a lot of memory. Hence, compressing images is essential for efficient image processing. You will later on see how Fourier analysis and Wavelet analysis can help us to compress an image significantly. There are also a few "computer scientific" tricks (for example entropy coding) to reduce the amount of data required to store an image.

3.4.1 Intensity image (gray scale image)

This is the equivalent to a "gray scale image" [13] and this is the image we are using in our project. It represents an image as a matrix where every element has a value corresponding to how bright/dark the pixel at the corresponding position should be colored. There are two ways to represent the number that represents the brightness of the pixel: The double class (or data type). This assigns a floating number ("a number with decimals") between 0 and 1 to each pixel. The value 0 corresponds to black and the value 1 corresponds to white. The other class is called uint8 which assigns an integer between 0 and 255 to represent the brightness of a pixel. The value 0 corresponds to black and 255 to white. The class uint8 only requires roughly 1/8 of the storage compared to the class double. On the other hand, many mathematical functions can only be applied to the double class.

3.4.2 Binary image

This image format also stores an image as a matrix but can only color a pixel black or white (and nothing in between). It assigns a 0 for black and a 1 for white.

3.4.3 Indexed image

This is a practical way of representing color images. An indexed image stores an image as two matrices. The first matrix has the same size as the image and one number for each pixel. The second matrix is called the color map and its size may be different from the image. The numbers in the first matrix is an instruction of what number to use in the color map matrix.

3.4.4 RGB image

This is another format for color images. It represents an image with three matrices of sizes matching the image format. Each matrix corresponds to one of the colors red, green or blue and gives an instruction of how much of each of these colors a certain pixel.

Chapter 4

Texture Representation

Texture [4, 16] provides a rich source of information about the natural scene. For designers, a texture adds richness to a design. For computer scientists, a texture is attractive not only because it is an important component in image analysis for solving a wide range of applied recognition, segmentation and synthesis problems, but also it provides a key to understanding basic mechanisms that underlie human visual perception.

A fundamental goal of texture research in computer vision is to develop automated computational methods for retrieving visual information and understanding image content based on textural properties in images. The critical issues in realizing the goal include understanding human texture perception and deriving appropriate quantitative texture descriptions.

4.1 Texture Features extraction

In many machine vision and image processing algorithms, simplifying assumptions are made about the uniformity of intensities in local image regions. Morever, images of real objects often do not exhibit regions of uniform intensities. For example, the image of a wooden surface is not uniform but contains variations of intensities which form certain repeated patterns called visual texture. The patterns can be the result of physical surface properties such as roughness or oriented strands which often have a tactile quality, or they could be the result of reflectance differences such as the color on a surface.

Image texture [16], defined as a function of the spatial variation in pixel intensities (gray values), is useful in a variety of applications and has been a subject of intense study by many researchers. One immediate application of image texture is the recognition of image regions using texture properties. For example, based on textural properties, we can identify a variety of materials such as cotton canvas, straw matting, raffia, herringbone weave, and pressed calf leather. Texture is the most important visual cue in identifying these types of homogeneous regions. This is called *texture* *classification*. The goal of texture classification then is to produce a classification map of the input image where each uniform textured region is identified with the texture class it belongs to.

For optimum classification purposes, what concern us are the statistical techniques of characterization. This is because it is these techniques that result in computing texture properties. The most popular statistical representations of texture are:

- Tamura Texture
- Wavelet Transform
- Local binary pattern

4.1.1 Tamura Texture

By observing psychological studies in the human visual perception, Tamura explored the texture representation using computational approximations to the three main texture features of: coarseness, contrast, and directionality [16]. Each of these texture features are approximately computed using algorithms.

- Coarseness is the measure of granularity of an image, or average size of regions that have the same intensity.
- Contrast is the measure of vividness of the texture pattern. Therefore, the bigger the blocks that makes up the image, the higher the contrast. It is affected by the use of varying black and white intensities.
- Directionality is the measure of directions of the grey values within the image.

4.1.2 Wavelet Transform

Textures can be modeled as quasi-periodic patterns with spatial/frequency representation. The wavelet transform [18] transforms the image into a multi-scale representation with both spatial and frequency characteristics. This allows for effective multi-scale image analysis with lower computational cost. According to this transformation, a function, which can represent an image, a curve, signal etc., can be described in terms of a coarse level description in addition to others with details that range from broad to narrow scales.

Unlike the usage of sine functions to represent signals in Fourier transforms, in wavelet transform, we use functions known as wavelets. Wavelets are finite in time, yet the average value of a wavelet is zero. In a sense, a wavelet is a waveform that is bounded in both frequency and duration. While the Fourier transform converts a signal into a continuous series of sine waves, each of which is of constant frequency and amplitude and of infinite duration, most real-world signals (such as music or images) have a finite duration and abrupt changes in frequency. This account for the efficiency of wavelet transforms. This is because wavelet transforms convert a signal into a series of wavelets, which can be stored more efficiently due to finite time, and can be constructed with rough edges, thereby better approximating real-world signals [26]. Examples of wavelets are Coiflet, Morlet, Mexican Hat, Haar and Daubechies. Of these, Haar is the simplest and most widely used, while Daubechies have fractal structures and are vital for current wavelet applications [19]. These two are outlined below:

4.2 Local Binary Pattern

The local binary pattern operator is an image operator which transforms an image into an array or image of integer labels describing small-scale appearance of the image. These labels or their statistics, most commonly the histogram, are then used for further image analysis. The most widely used versions of the operator are designed for monochrome still images but it has been extended also for color images as well as videos and volumetric data. This chapter covers the different versions of the actual LBP operator in spatial domain [47, 48].

4.2.1 Basic LBP

The basic local binary pattern operator, introduced by Ojala et al.[47], was based on the assumption that texture has locally two complementary aspects, a pattern and its strength. In that work, the LBP was proposed as a two-level version of the texture unit [48] to describe the local textural patterns.

The original version of the local binary pattern operator works in a 3×3 pixel block of an image. The pixels in this block are thresholded by its center pixel value, multiplied by powers of two and then summed to obtain a label for the center pixel. As the neighborhood consists of 8 pixels, a total of 2^8 = 256 different labels can be obtained depending on the relative gray values of the center and the pixels in the neighborhood. See Fig. 4.1 for an illustration of the basic LBP operator. An example of an LBP image and histogram are shown in Fig. 4.2.

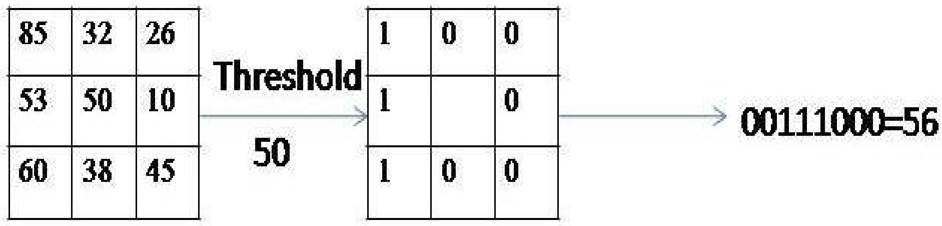


Figure 4.1 basic LBP operator

4.2.2 Derivation of the Generic LBP Operator

Several years after its original publication, the local binary pattern operator was presented in a more generic revised form by Ojala et al. [48]. In contrast to the basic

Fig. 4.2 Example of an input image, the corresponding LBP image

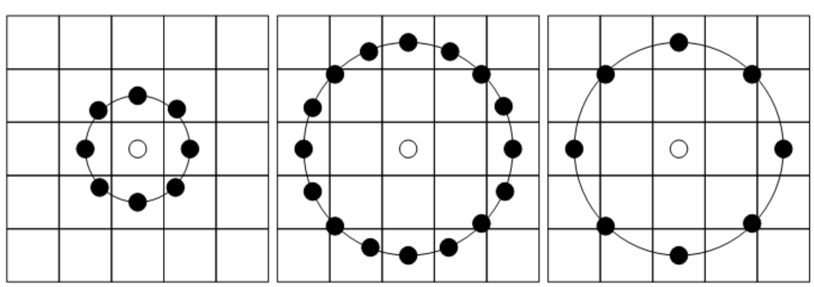


Fig: - 4.3 the circular (8, 1), (16, 2) and (8, 2) neighborhoods. The pixel values are bilinearly interpolated whenever the sampling point is not in the center of a pixel.

LBP using 8 pixels in a 3 × 3 pixel block, this generic formulation of the operator puts no limitations to the size of the neighborhood or to the number of sampling points. The derivation of the generic LBP presented below follows that of [47, 48].

Consider a monochrome image I (x, y) and let g_c denote the gray level of an arbitrary pixel (x, y), i.e. g_c = I(x, y). Moreover, let g_p denote the gray value of a sampling point in an evenly spaced circular neighborhood of P sampling points and radius R around point (x, y):

$$g_p = I(x, y), \quad p=0, \ldots\ldots\ldots, p-1 \text{ and}$$

$$x_p = x + R\cos(\frac{2\pi p}{P}),$$

$$y_p = y - R\sin(\frac{2\pi p}{P}).$$

See Fig. 4.2 for examples of local circular neighborhoods.

Assuming that the local texture of the image I(x, y) is characterized by the joint distribution of gray values of P + 1 (P > 0) pixels:

$$T = t\,(g_{c,} g_0, g_1, \ldots\ldots\ldots\ldots\ldots\ldots g_{p-1}) \qquad 4.1$$

without loss of information, the center pixel value can be subtracted from the neighborhood:

$$T = t\,(g_{c\,,} g_0 - g_{c,}, g_1 \text{-} g_{c,}, \ldots\ldots\ldots\ldots\ldots\ldots g_{p-1} - g_{c,}) \qquad 4.2$$

In the next step the joint distribution is approximated by assuming the center pixel to be statistically independent of the differences, which allows for factorization of the distribution:

$$T \approx t\,(g_{c\,,})t(g_0 - g_{c,}, g_1 \text{-} g_{c,}, \ldots\ldots\ldots\ldots\ldots\ldots g_{p-1} - g_{c,}) \qquad 4.3$$

Now the first factor t ($g_{c\,,}$) is the intensity distribution over I(x,y). From the point of view of analyzing local textural patterns, it contains no useful information. Instead the joint distribution of differences

$$t(g_0 - g_{c,}, g_1 \text{-} g_{c,}, \ldots\ldots\ \ldots\ldots\ldots\ldots g_{p-1} - g_{c,}) \qquad 4.4$$

can be used to model the local texture. Morever, reliable estimation of this multi dimensional distribution from image data can be difficult. One solution to this problem, proposed by Ojala et al. in [47], is to apply vector quantization. They used learning vector quantization with a codebook of 384 code words to reduce the dimensionality of the high dimensional feature space. The indices of the 384 code words correspond to the 384 bins in the histogram. Thus, this powerful operator based on signed gray-level differences can be regarded as a text on operator, resembling some more recent methods based on image patch exemplars (e.g. [48]).

The learning vector quantization based approach still has certain unfortunate properties that make its use difficult. First, the differences g_p-$g_c$are invariant to changes of the mean gray value of the image but not to other changes in gray levels. Second, in order to use it for texture classification the codebook must be trained similar to the other texton-based methods. In order to alleviate these challenges, only the signs of the differences are considered:

$$t(s(g_0 - g_{c,}, s(g_1 - g_{c,}), \ldots\ldots \ldots\ldots\ldots s(g_{p-1} - g_{c,)}) \qquad 4.5$$

where s(z) is the thresholding (step) function

$$s(z) = \begin{cases} 1, & z \geq 0 \\ 0, & z < 0 \end{cases}$$

The generic local binary pattern operator is derived from this joint distribution. As in the case of basic LBP, it is obtained by summing the thresholded differences weighted by powers of two. The $LBP_{P,R}$ operator is defined as

$$LBP_{P,R}(x_c\ ,y_c) = \sum_{i=0}^{p-1} s(g_p\text{-}g_c)2^p \qquad 4.6$$

In practice, Eq. 2.10 means that the signs of the differences in a neighborhood are interpreted as a P -bit binary number, resulting in 2P distinct values for the LBP code. The local gray-scale distribution, i.e. texture, can thus be approximately described with a 2P -bin discrete distribution of LBP codes:

$$T = t(LBP_{P,R}(x_c\ ,y_c)) \qquad 4.7$$

In calculating the $LBP_{P,R}$ distribution (feature vector) for a given N × M image sample ($x_c \in \{0,...,N-1\}$, $y_c \in \{0,...,M-1\}$), the central part is only considered because a sufficiently large neighborhood cannot be used on the borders. The LBP code is calculated for each pixel in the cropped portion of the image, and the distribution of the codes is used as a feature vector, denoted by S:

$$S = t(LBP_{P,R}(x_c, y_c)) \qquad 4.8$$

$x \in \{ \lceil R \rceil,\ldots\ldots,N-1-\lceil R \rceil\}$, $y \in \{\lceil R \rceil,\ldots\ldots\ldots\ldots\ldots M-1-\lceil R \rceil\}$.

The original LBP (Fig. 4.1) is very similar to$LBP_{8,1}$, with two differences. First, the neighborhood in the general definition is indexed circularly, making it easier to derive rotation invariant texture descriptors. Second, the diagonal pixels in the 3×3 neighborhood are interpolated in $LBP_{8,1}$.

4.2.3 Mappings of the LBP Labels: Uniform Patterns

In many texture analysis applications it is desirable to have features that are invariant or robust to rotations of the input image. As the $LBP_{P,R}$ patterns are obtained by circularly sampling around the center pixel, rotation of the input image has two effects: each local neighborhood is rotated into other pixel location, and within each neighborhood, the sampling points on the circle surrounding the center point are rotated into a different orientation.

Another extension to the original operator uses so called uniform patterns [47]. For this, a uniformity measure of a pattern is used: U (“pattern”) is the number of bitwise transitions from 0 to 1 or vice versa when the bit pattern is considered circular. A local binary pattern is called uniform if its uniformity measure is at most 2.For example, the patterns 00000000 (0 transitions), 01110000 (2 transitions) and 11001111 (2 transitions) are uniform whereas the patterns 11001001 (4 transitions) and 01010011 (6 transitions) are not. In uniform LBP mapping there is a separate output label for each uniform pattern and all the non-uniform patterns are assigned to a single label. Thus, the number of different output labels for mapping for patterns

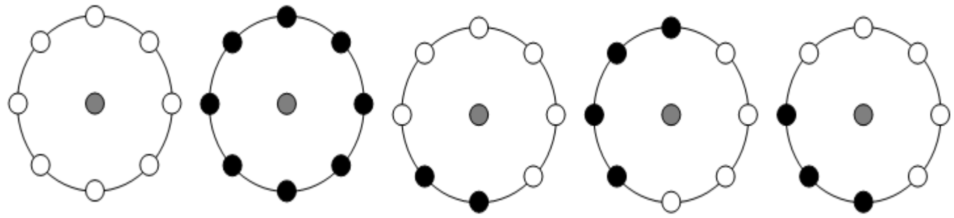


Fig. 4.4 Different texture primitives detected by the LBP

of P bits is P(P − 1) + 3. For instance, the uniform mapping produces 59 output labels for neighborhoods of 8 sampling points, and 243 labels for neighborhoods of 16 sampling points.

The reasons for omitting the non-uniform patterns are twofold. First, most of the local binary patterns in natural images are uniform. Ojala et al. noticed that in their experiments with texture images, uniform patterns account for a bit less than 90% of all patterns when using the (8, 1) neighborhood and for around 70% in the (16, 2) neighborhood. In experiments with images [48] it was found that 90.6% of the patterns in the (8, 1) neighborhood and 85.2% of the patterns in the (8, 2) neighborhood are uniform.

The second reason for considering uniform patterns is the statistical robustness. Using uniform patterns instead of all the possible patterns has produced better recognition results in many applications. On one hand, there are indications that uniform patterns themselves are more stable, i.e. less prone to noise and on the other hand, considering only uniform patterns makes the number of possible LBP labels significantly lower and reliable estimation of their distribution requires fewer samples.

The uniform patterns allow seeing the LBP method as a unifying approach to the traditionally divergent statistical and structural models of texture analysis [45]. Each pixel is labeled with the code of the texture primitive that best matches the local neighborhood. Thus each LBP code can be regarded as a micro-texton. Local primitives detected by the LBP include spots, flat areas, edges; edge ends, and curves and so on. Some examples are shown in Fig. 4.3 with the $LBP_{8,R}$ operator. In the figure, ones are represented as black circles, and zeros are white.

The combination of the structural and statistical approaches stems from the fact that the distribution of micro-textons can be seen as statistical placement rules. The LBP distribution therefore has both of the properties of a structural analysis method: texture primitives and placement rules. On the other hand, the distribution is just a statistic of a non-linearly filtered image, clearly making the method a statistical one. For these reasons, the LBP distribution can be successfully used in recognizing a wide variety of different textures, to which statistical and structural methods have normally been applied separately.

4.3 image descriptor

In each block of images, histogram is independently obtained based on LBP operator and its variants. If the image is divided into N blocks, N histograms are calculated. Finally, N histograms are concatenated to yield the image descriptor. In the same way, 2D LBP histogram, which is composed of two of LBP and its 3 variants, 3D histogram (MGST, AP, Shape) proposed in the earlier work by the author [43], and 4D histogram, which combined proposed 3D histogram and LBP, are also extracted for comparison. Additionally, 3D histogram which combined three of MGST, AP [42], Shape descriptor based on FAST corner detector [43] and LBP is also extracted. MGST measures symmetrical similarity in local region and AP measures directionality of the pixel intensity change. Both of them are robust to rotation. The proposed shape descriptor in the earlier work by the author was refined to have 16 levels instead of 32 levels.

4.4 Matching

The similarity of image descriptor formed by combining N histograms was measured by several dissimilarity measures: Bhattacharyya distance, Histogram intersection, log-likelihood, Chi square, and correlation distance. F_j and F_j (i) denotes the image descriptor of j-th image and its i^{th} value, respectively. E() and S() denotes averaging operator, and standard deviation operator respectively.

4.5 Minimum distance classifiers

The minimum distance classifiers, as the name suggests, classify the data on the basis of the distance between the data points in the feature space. It classifies in such a manner, so as to minimize the distance between the data and the class in multi-feature space. The index of similarity here is the distance so that the minimum distance is identical to the maximum similarity. The most common distances often used in this procedure are:

- Bhattacharya distance

$$d(F_1, F_2) = \sqrt{1 - \sum_i \sqrt{\frac{F_1(\mathrm{i}) * F_2(\mathrm{i})}{E(F_1) * E(F_2)}}}$$

- Chi square statistics

$$d(F_1, F_2) = \sum_i \frac{((\mathrm{F}_1(\mathrm{i}) - F_2(\mathrm{i}))^2}{((\mathrm{F}_1(\mathrm{i}) + F_2(\mathrm{i}))^2}$$

- Histogram intersection

$$d(F_1, F_2) = \sum_i \min\,(F_1(i), F_2(i))$$

- Correlation distance

$$d(F_1, F_2) = \frac{E((F_1 - E(F_1) * (F_2 - E(F_2))}{S(F_1) * S(F_2)}$$

- Log- likelihood statistics

$$d(F_1, F_2) = \sum F_1(i) * \log\,(F_2(i))$$

The weight for B_i , $w_i$can be applied to the dissimilarity results of histogram in B_i to form the final dissimilarity. It is because that each block of the image has not valuable information equivalently. Query images are classified into the person with the lowest dissimilarity according to matching results.

Chapter 5

Experiment and Results

In this chapter we present retrieval system that supports color and texture features. In our system image is represented by set of features that are extracted by using color and texture properties of image which are described in this chapter. These extracted features are then compared with the stored images features in the database. The retrieved images are ranked and top results are then displayed.

5.1 Database

The WANG database is a subset of 1,000 images of the Corel stock photo database which have been manually selected and which form 10 classes of 100 images each. The WANG database can be considered similar to common stock photo retrieval tasks with several images from each category and a potential user having an image from a particular category and looking for similar images which have e.g. cheaper royalties or which have not been used by other media. The 10 classes are used for relevance estimation: given a query image, it is assumed that the user is searching for images from the same class, and therefore the remaining 99 images from the same class are considered relevant and the images from all other classes are considered irrelevant. Database is available at this link http://wang.ist.psu.edu/docs/related/

Figure 5.1: sample images of corel stock photo database

5.2 Quadratic Distance Metric

Color histogram is widely used for color based image retrieval in content-based image retrieval. A color histogram describes the global color distribution of an image. It is very easy to compute and is insensitive to small changes in viewing positions. The equation we used in deriving the distance between two color histograms is the quadratic distance metric:

$$d^2(Q,I) = \left(H_Q - H_I\right)^t A(H_Q - H_I)$$

The equation consists of three terms. The derivation of each of these terms will be explained in the following sections. The first term consists of the difference between two color histograms; or more precisely the difference in the number of pixels in each bin. This term is obviously a vector since it consists of one row. The number of columns in this vector is the number of bins in a histogram. The third term is the transpose of that vector. The middle term is the similarity matrix. The final result **d** represents the color distance between two images. The closer the distance is to zero the closer the images are in color similarity. The further the distance from zero the less similar the images are in color similarity.

5.2.1 Color Extraction & Matching:

Using the color feature extraction algorithm described below, where the histograms of the query image and the images in the database are compared using the Quadratic Distance Metric.

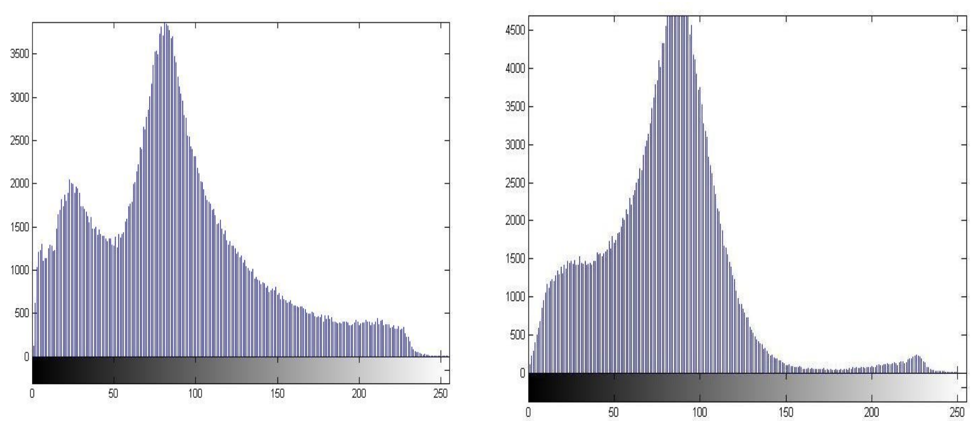


Figure: 5.2 histogram of images in database and of test image

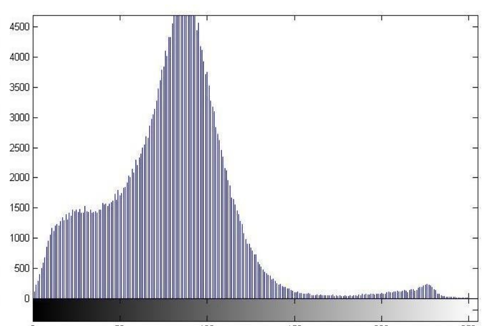
5.3 Texture Features extraction using LBP:

One of the best performing texture descriptors and widely used in various applications is local binary pattern. It has proven to be highly discriminative and because its invariance to monotonic gray level changes and computational efficiency, make it suitable for demanding image analysis tasks.

5.3.1Pre processing:

RGB and indexed images carry high values that require more computation time. Hence, the images are converted to grayscale in order to reduce the vast spectrum of indexed images or the 3D components of RGB to a 2D component carrying values between 0 and 255 (containing the end points). This process promises reduction in the computation time and power required for extracting features from an image. The resulting image undergoes histogram equalization in order to enhance contrast of values of an image by generating its flat histogram. Preprocessing applied to the image shown in figures.

(i)



(ii) (iii)

Figure 5.3: (i) Image (ii) RGB to gray image (iii) Its Histogram

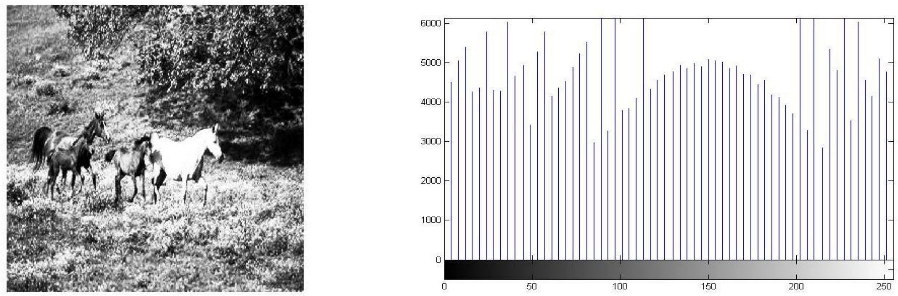


Figure 5.4 (iv) resized image and (v) equalized histogram

5.3.2 LBP Algorithm:

We are using LBP for texture features extraction of images. Representation of image based on LBP can be summarized as:

Image→several regions→extract and concatenate LBP feature distributions into a vector which used as a image descriptor as shown in figure.

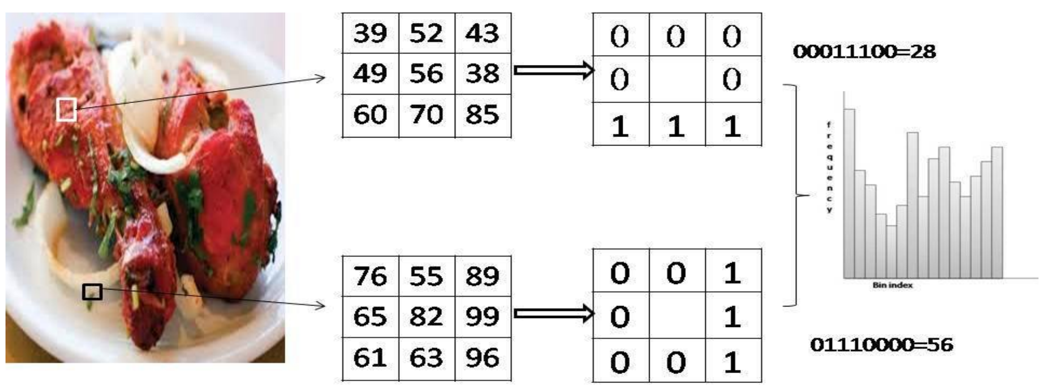


Figure: 5.5 Illustration of LBP. Typically the binary codes obtained by local thresholding are transformed into decimal code.

5.3.3 Similarity measurement for texture

We are using Euclidean Distance based k-mean nearest neighbour classifier for comparing distance between texture features of query image and database images.

The k-means algorithm [11] is a fast iterative algorithm that has been used in many clustering applications. It finds locally optimal solutions with respect to the clustering error. Let us consider a data set X= $\{x_1, x_2, x_3, \dots \dots \dots \dots x_n\}$ X $\in R^d$ where d is the dimension. The K-clustering problem aims at partitioning this data set into K independent clusters $\{C_{1,} C_2, C_3, \dots \dots \dots \dots C_k\}$, such that squared Euclidean distances between each data point x_i and the centroid m_c (cluster center) of the subset C_K is minimized. This criterion is called clustering error and depends on the cluster centers { $m_1 m_2 m_3 \dots \dots \dots .. m_K$}

E $(m_1 m_2 m_3 \dots \dots \dots .. m_K) = \sum_{i=1}^{M} \sum_{j=1}^{N} I(x_i \in C_j) ||x_i - m_j||^2$

Where I(X) =1 if X is true and 0 otherwise.

5.3.4 K-Means clustering algorithm

- Initially, the number of clusters must be known, or chosen, to be K say.
- The initial step is the choose a set of K instances as centres of the clusters. Often chosen such that the points are mutually “farthest apart”, in some way.
- Next, the algorithm considers each instance and assigns it to the cluster which is closest.
- The cluster centroids are recalculated either after each instance assignment, or after the whole cycle of re-assignments.
- This process is iterated.

5.4 Evaluation metric

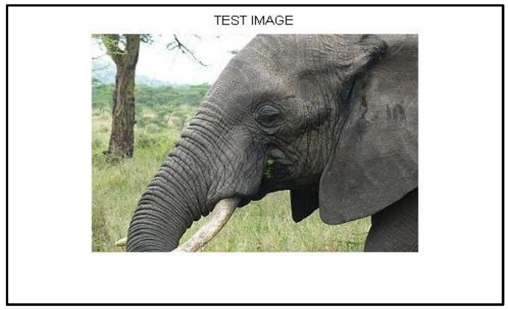
To evaluate CBIR, several performance evaluation measures have been proposed. In this report, we exploit the precision P and recall R to evaluate the proposed method:

$$P = \frac{\text{Number of relevant images retrived}}{\text{Total number of images retrived}}$$

$$\mathrm{R} = \frac{\text{Number of relevant images retrived}}{\text{Total number of relevant images in database}}$$

5.5 Results: we are showing here a rough demonstration of our project application with the help of this example:

- Here we are taking a test image that is not in our database as shown in figure.



(i)

- Using the *feature extraction algorithm* described above corresponding top 10 retrieved images are shown in figure.

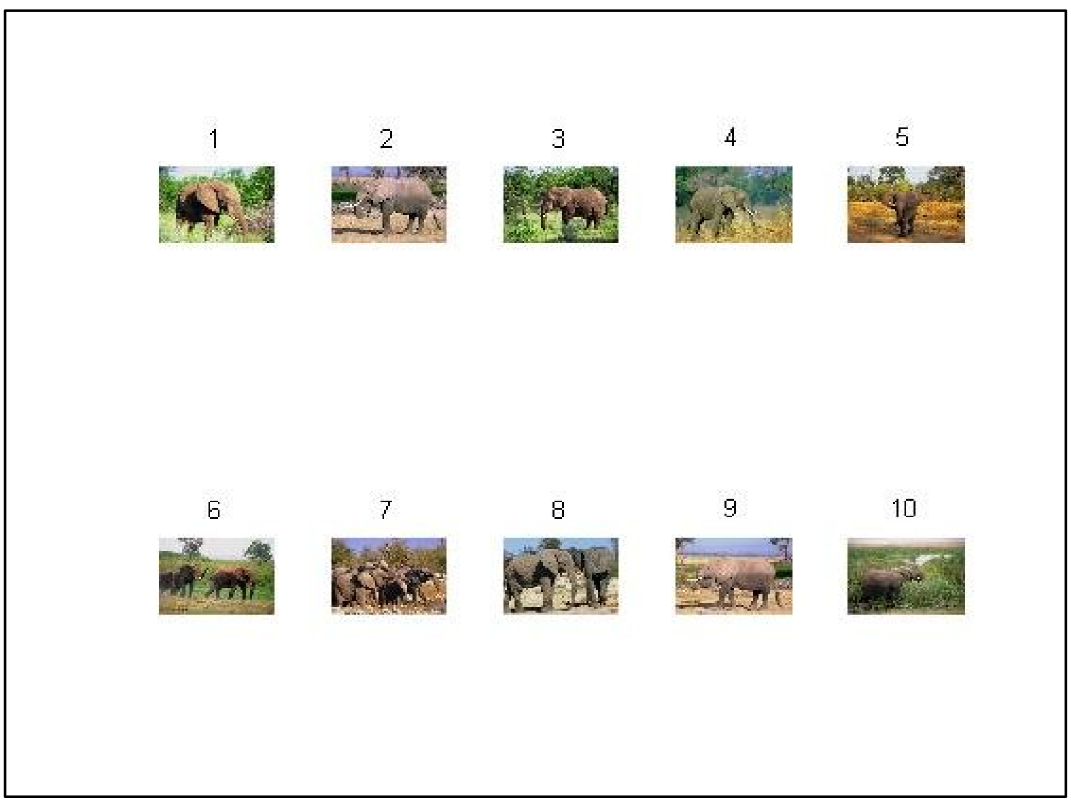


Figure: 5.6 (i) Test image (ii) Top 10 retrieved image.

- This table is showing the Euclidian distance between test image with relevant retrieved images.

Table 1

File Name	*Distance measurement*
45.jpg	2.6490
49.jpg	3.3804
29.jpg	4.3435
31.jpg	5.0800
11.jpg	5.9940
41.jpg	6.6100
51.jpg	6.9638
91.jpg	7.0813
81.jpg	7.1060
51.jpg	7.1958
……	…..

Belongs to elephant class

Total time in searching 9.337333

Recall= 62.85%

Precision = 78.45%

Chapter 6

Conclusion

In this report, content based image retrieval system is presented. Systems using CBIR retrieve images based on visual features such as color, and texture. The application performs a simple color-based search in an image database for an input query image, using color histograms. It then compares the color histograms of different images using the *Quadratic Distance Equation*. Further to improve the searching process, the application performs a texture-based search in the color results, using *local binary pattern* as an image texture descriptor. It then compares the texture features obtained using the Euclidean distance based *k-mean nearest neighbor classifier*.

In image processing field CBIR is still developing science. Image feature descriptor and feature extraction techniques are becoming more and more advanced. This development promises an immense range of future applications using CBIR.

References

1. Ricardo da Silva Torres, Alexandre Xavier Falcao,"Content-Based Image Retrieval: Theory and Applications" RITA • Volume XIII • Número 2 • 2006
2. Arnold W. M. Smeulders, Marcel worring, Member, "content based image retrieval at the end of the early years". IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 12, December 2000
3. Szabolcs Sergyán Budapest Tech, "Color Content-based Image Classification", 5th Slovakian-Hungarian Joint Symposium on Applied Machine Intelligence and Informatics
   January 25-26, 2007.
4. Bharathi P.T, Dr. P. Subashini, "Texture Based Color Segmentation For Infrared River Ice Images Using K-Means Clustering", 2013 International Conference on Signal Processing, Image Processing and Pattern Recognition
5. Abdelhamid Abdesselam, "Texture Image Retrieval Using Fourier Transform", International Conference On Communication, Computer And Power (Icccp'2009).
6. Matthew Partridge and Marwan Jabri, "Robust Principal Component Analysis", School of Electrical and Information Engineering University of Sydney.
7. W. Y. Ma and B. S. Manjunath, "NeTra: A Toolbox for Navigating Large Image Databases", Multimedia Systems ©Springer-Verlag 1999
8. M. J. Swain, and D. H. Ballard, "Color indexing", International Journal of Computer Vision, Vol. 7, No. 1, pp.11-32, 1991.
9. J. R. Smith and S.-F. Chang, " Querying by color regions using the VisualSEEk content-based visual query system", In M. T. Maybury, editor, Intelligent Multimedia Information Retrieval. AAAI Press, 1997.
10. G J McLachlan, "Mahalanobis Distance", General Article.
11. Paul Lammertsma, "K-nearest-neighbor algorithm" available at paul.luminos.nl/download/document/knn.pdf
12. Timothy K. Shih,Jiung - Yao Huang, Ching - Sheng Wang, and Chuan - Ho Kao, "An Intelligent Content-based Image Retrieval System based on Color, Shape and Spatial Relations Tamkang University Tamsui, Taiwan, R.O.C.

13. Timo Ojala, Matti Pietikäinen and T opi Mäenpää, “Gray Scale and Rotation Invariant T exture Classification with Local Binary Patterns”, Machine Vision and Media Processing Unit Infotech Oulu, University of Oulu
14. Pankaj Kumar, Kuntal Sengupta, Adrian Lee, “A Comparative Study of Different Color Spaces for Foreground and Shadow Detection for Traffic Monitoring System” The IEEE 5fi International Conference on Intelligent Transportation Systems 3 - 6 September 2002, Singapore.
15. Theory of color space.
16. Irene Epifanio and Guillermo Ayala, “A Random Set View of Texture Classification”, IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 11, NO. 8, AUGUST 2002.
17. Tom´ aˇ s Majtner, David Svoboda, “Extension of Tamura Texture Features for 3D Fluorescence Microscopy”, 2012 Second Joint 3DIM/3DPVT Conference: 3D Imaging, Modeling, Processing, Visualization & Transmission.
18. Chun-Lin, Liu, “A Tutorial of the Wavelet Transform”, February 23, 2010.
19. Masoud Mazloom, Shohreh Kasaei, “Construction and Application of SVM Model and Wavelet-PCA for Face recognition”, 2009 Second International Conference on Computer and Electrical Engineering.
20. R.Raja, S.Md.Mansoor Roomi, D.KalaiyarasSemantic, “Modeling of Natural scenes by Local Binary Pattern”, 978-1-4673-2322-2/12/$31.00 ©2012 IEEE.
21. . Dandge S.S. and Bodkhe A.P. “Content Based Image Retrieval System”, Journal of Signal and Image Processing ISSN: 0976-8882 & E-ISSN: 0976-8890, Volume 3, Issue 2, 2012,
22. Li Liu, Paul Fieguth, Gangyao Kuang, “Generalized Local Binary Patterns for Texture Classification”, A bibliography of LBP-related research can be found at the Oulu University website: http://www.cse.oulu.fi/MVG/LBP_Bibliography/ © 2011.
23. pietikainen, M. hadid, Ziao G., Ahonen T., “Local Binary Patterns for Still Images”, springer 2011 available at http://www.springer.com/978-0-85729-747-1
24. Taskeed Jabid, Md. Hasanul Kabir, Oksam Chae, “Local Directional Pattern (LDP) – A Robust Image Descriptor for Object Recognition” 2010 Seventh

IEEE International Conference on Advanced Video and Signal Based Surveillance.

25. P. V. N. Reddy & K. Satya Prasad, “Local Derivative Patterns and their Magnitudes for Content Based Image Retrieval”, International Journal of Computer Science & Informatics, Volume-I, Issue-II, 2011
26. Waqas Rasheed, Youngeun An, “Image Retrieval using Maximum Frequency of
Local Histogram based Color Correlogram”, 2008 International Conference on Multimedia and Ubiquitous Engineering.
27. E. de Ves, A. Ruedin, D. Acevedo, X.Benavent, and L. Seijas, “A New Wavelet-Based Texture Descriptor forImage Retrieval”, Springer-Verlag Berlin Heidelberg 2007.
28. Maenpaa Topi, Pietikainen Matti, Ojala Timo, “Texture Classification by Multi-Predicate Local Binary Pattern Operators”, © 2000 IEEE.
29. Yanyan Gao, Honggang Zhang, Jun Guo, “Multiple Features-Based Image Retrieval” Proceedings of IEEE IC-BNMT2011.
30. “Content-based Image Retrieval Using Gabor-Zernike Features” The 18th International Conference on Pattern Recognition (ICPR'06)©2006 IEEE.
31. Taskeed Jabid, Md. Hasanul Kabir, and Oksam Chae, “Robust Facial Expression Recognition Based on Local Directional Pattern” ETRI Journal, Volume 32, Number 5, October 2010.
32. . Dengsheng Zhangn, Md. Monirul Islam, Guojun Lu “A review on automatic image annotation techniques”, & 2011 Elsevier Ltd. All rights reserved
33. Suli Zhang, Xin Pan, “A novel Text Classification Based on Mahalanobis distance”, 978-1-61284-840-2/11/$26.00 ©2011IEEE.
34. Xiannong Meng, Rishav Chakravarti, “A Study of Color Histogram Based Image Retrieval”, 2009 Sixth International Conference on Information Technology: New Generations
35. Zhenhua Zhang, Wenhui Li, Bo Li, “An Improving Technique of Color Histogram in Segmentation-based Image Retrieval”, 2009 Fifth International Conference on Information Assurance and Security.
36. Ramin Zabih, “IEEE Transactions on Pattern Analysis and Machine Intelligence” VOL. 32, NO. 1, JANUARY 2010.

37. Douglas R. Heisterkamp Jing Peng H. K. Dai " Feature Relevance Learning with Query Shifting for Content-Based Image Retrieval", 0-7695-0750-6/00 $10.00 0 2000 IEEE
38. Wang Database for content based image retrieval system available at http://savvash.blogspot.in/2008/12/benchmark-databases-for-cbir.html
39. MATLAB help
40. Hybrid color spaces applied to image database, J. D. Rugna, P. Colantoni,and N.Boukala,, vol. 5304, pp. 254{264, Electronic Imaging, SPIE, 2004
41. Won-Keun Yang, Ayoung Cho, Dong-Seok Jeong, and Weon-Geun Oh, "Image Description and Matching Scheme for Identical Image Searching", Conf. on Advanced Service Computing, IEEE, pp.669-674, 2009
42. Keun-Dong Lee, Seon-Tae Kim, Sung-Kwang Je, and Weon-Geun Oh, "Face recognition for personal image management system," in Proceedings of the 16th Korea-Japan Joint Workshop on Frontiers of Computer Vision, 2009
43. Varma, M., Zisserman, A.: " A statistical approach to materials classification using image patchexemplars." IEEE Trans. Pattern Anal. Mach. Intell.31, 2032–2047 (2009)
44. Ahonen, T., Hadid, A., Pietikäinen, M.: " Face description with local binary patterns: Application to face recognition". IEEE Trans. Pattern Anal. Mach. Intell. 28(12), 2037–2041 (2006)
45. Ojala, T., Valkealahti, K., Oja, E., Pietikäinen, M.: "Texture discrimination with multidimensional distributions of signed gray-level differences". Pattern Recognit. 34(3), 727–739 (2001)
46. Wang, L., He, D.C.: "Texture classification using texture spectrum. Pattern Recognit". 23, 905–910 (1990)
47. Ojala, T., Pietikäinen, M., Harwood, D.: "A comparative study of texture measures with classification based on feature distributions". Pattern Recognit. 29(1), 51–59 (1996)

48. Ojala, T., Pietikäinen, M., Mäenpää, T.: "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns". IEEE Trans. Pattern Anal. Mach. Intell. 24(7),971–987 (2002)

49. Mäenpää, T., Pietikäinen, M.: "Texture analysis with local binary patterns". In: Chen, C.H.,Wang, P.S.P. (eds.) Handbook of Pattern Recognition and Computer Vision, 3rd edn., pp.197–216. World Scientific, Singapore (2005)

50. James Z. Wang,Senior Member, "Real-World Image Annotation and Retrieval:An Introduction to the Special Section", IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 30, NO. 11, NOVEMBER 2008

51. G. Rafiee, S.S. Dlay, and W.L. Woo, "A Review of Content-Based Image Retrieval", 978-1-86135-369-6/101$25.00 ©2010 IEEE.

52. R. Datta, D. Joshi, J. Li, and J. Z. Wang, "Image Retrieval: Ideas, Influences, and Trends of the New Age," ACM Computing Survey, vol. 40, pp. 5: 1-60, 2008

Appendix A

CBIR	(Content Based Image Retrieval)- The process of retrieving images based on visual features such as texture and color.
Color distance	The degree of similarity between two color histograms represented by a numerical distance. The closer the distance is to zero the more similar two images are in terms of color. The further away the distance is from zero, the less similar.
Color histogram	A histogram that represents the color information of an image. The x-axis represents all the different colors found in the image. Each specific color is referred to as a bin. The y-axis represents the number of pixels in each bin.
Color space	The three dimensional space in which color is defined.
Euclidean distance	Equation used to compare average intensities of pixels.
Global color histogram	The color histogram of the whole image.
HSV	Hue Saturation Value color space.
Local color histogram	The color histogram of a subsection of an image.
Quadratic metric	Equation used to calculate the color distance. Consists of three terms, color histogram difference, transpose of color histogram difference, and similarity matrix.
Quantization	Reducing the number of color bins in a color map by taking colors that are very similar to each other and putting them in the same bin.
QBIC	CBIR system developed by IBM.
RGB	Red Green Blue color space
Similarity matrix	Matrix that denotes the similarity between two sets of data. This is identified by the diagonal of the matrix. The closer the sets of data, the closer the diagonal is to one. The farther the sets of data, the farther the diagonal is from one.

Appendix B

An introduction to image processing

MATLAB is a high performance language for technical computing. It integrates computation, visualization and programming in an easy to use environment where problems and solutions are expresses in familiar mathematical notation. It is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations in a fraction of the time it would take to write a program in a scalar non interactive language such as C, C++ or JAVA.

Image formats supported by MATLAB

The following image formats are supported by MATLAB:

- JPEG
- PCX
- BMP
- TIFF
- HDF
- XWB

Working formats in MATLAB

If an image is stored as a JPEG-image on your disc we first read it into MATLAB. Morever, in order to start working with an image, for example perform a wavelet transform on the image, we must convert it into a different format.

Intensity image (gray scale image)

It represents an image as a matrix where every element has a value corresponding to how bright/dark the pixel at the corresponding position should be colored. There are two ways to represent the number that represents the brightness of the pixel: The double class (or data type). This assigns a floating number ("a number with decimals") between 0 and 1 to each pixel. The value 0 corresponds to black and the value 1 corresponds to white. The other class is called uint8 which assigns an integer between 0 and 255 to represent the brightness of a pixel.

Binary image

This image format also stores an image as a matrix but can only color a pixel black or white (and nothing in between). It assigns a 0 for black and a 1 for white.

Indexed image

This is a practical way of representing color images. An indexed image stores an image as two matrices. The first matrix has the same size as the image and one number for each pixel. The second matrix is called the color map and its size may be different from the image. The numbers in the first matrix is an instruction of what number to use in the color map matrix.

RGB image

This is another format for color images. It represents an image with three matrices of sizes matching the image format. Each matrix corresponds to one of the colors red, green or blue and gives an instruction of how much of each of these colors a certain pixel should use.

Appendix C

Overview of MATLAB Environment

MATLAB is a high-level technical computing language and interactive environment for algorithm development, data visualization, data analysis, and numeric computation. Using the MATLAB product, you can solve technical computing problems faster than with traditional programming languages, such as C, C++, and Fortran.

You can use MATLAB in a wide range of applications, including signal and image processing, communications, control design, test and measurement, financial modeling and analysis, and computational biology. Add-on toolboxes (collections of special-purpose MATLAB functions, available separately) extend the MATLAB environment to solve particular classes of problems in these application areas.

MATLAB provides a number of features for documenting and sharing your work. You can integrate your MATLAB code with other languages and applications, and distribute your MATLAB algorithms and applications. Features include:

- High-level language for technical computing
- Development environment for managing code, files, and data
- Interactive tools for iterative exploration, design, and problem solving
- Mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization, and numerical integration
- 2-D and 3-D graphics functions for visualizing data
- Tools for building custom graphical user interfaces
- Functions for integrating MATLAB based algorithms with external applications and languages, such as C, C++, Fortran, Java™, COM, and Microsoft® Excel®

The MATLAB System:

The MATLAB system consists of these main parts:

Desktop Tools and Development Environment

This part of MATLAB is the set of tools and facilities that help you use and become more productive with MATLAB functions and files. Many of these tools are graphical user interfaces. It includes: the MATLAB desktop and Command Window, an editor and debugger, a code analyzer, and browsers for viewing help, the workspace, and folders.

Mathematical Function Library

This library is a vast collection of computational algorithms ranging from elementary functions, like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix eigen values, Bessel functions, and fast Fourier transforms.

The Language

The MATLAB language is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick programs you do not intend to reuse. You can also do "programming in the large" to create complex application programs intended for reuse.

Graphics

MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It includes high-level functions for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level functions that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications.

External Interfaces

The external interfaces library allows you to write C/C++ and Fortran programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), for calling MATLAB as a computational engine, and for reading and writing MAT-files.